

Nonnegative Matrix *Underapproximation* for Robust Multiple Model Fitting*

Mariano Tepper Guillermo Sapiro
Department of Electrical and Computer Engineering, Duke University
{mariano.tepper, guillermo.sapiro}@duke.edu

Abstract

In this work, we introduce a highly efficient algorithm to address the nonnegative matrix underapproximation (NMU) problem, i.e., nonnegative matrix factorization (NMF) with an additional underapproximation constraint. NMU results are interesting as, compared to traditional NMF, they present additional sparsity and part-based behavior, explaining unique data features. To show an example of these features in practice, we first present an application to the analysis of climate data. We then present an NMU-based algorithm to robustly fit multiple parametric models to a dataset. The proposed approach delivers state-of-the-art results for the estimation of multiple fundamental matrices and homographies, outperforming other alternatives in the literature and exemplifying the use of efficient NMU computations.

1. Introduction

Nonnegative Matrix Factorization (NMF) [13] has been successfully applied as a data analysis technique, mainly because its part-based representation and sparsity. Unfortunately, the problem of finding a rank- r NMF of a nonnegative matrix \mathbf{A} is NP-hard [20]. However, a rank-one NMF can be obtained in polynomial time. On the one hand, the rank-one singular value decomposition (SVD) of \mathbf{A} is an optimal solution to the problem

$$\min_{\text{rank}(\mathbf{Z})=1} \|\mathbf{A} - \mathbf{Z}\|_F^2. \quad (1)$$

On the other hand, the Perron-Frobenius theorem implies that the SVD factors with the largest singular value of a nonnegative matrix are themselves nonnegative [12, Chapter 8]. Hence, the SVD provides an optimal rank-one NMF.

Following a standard approach in the literature [14, 21] we might be tempted to iterate two steps until the desired rank is reached:

1. compute the rank-one SVD $\mathbf{su}\mathbf{v}^T$ of \mathbf{A} , and
2. set $\mathbf{A} = \mathbf{A} - \mathbf{su}\mathbf{v}^T$.

*Work partially supported by ONR, NGA, ARO, and NSF.

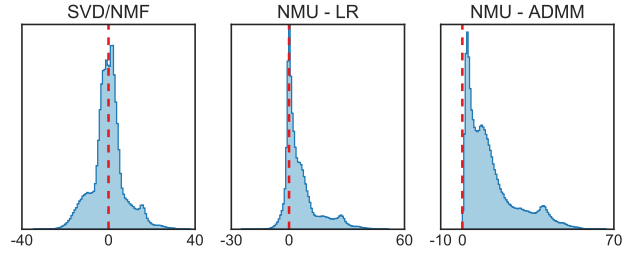


Figure 1. In this example we show the histogram of the values of $\mathbf{A} - \mathbf{uv}^T$. NMF (which coincides with the SVD in the rank-one case) is not expected to yield any particular pattern. NMU must yield nonnegative values. Clearly, the Lagrangian relaxation method (LR), reported as the state-of-the-art in the literature until now, does not correctly enforce the underapproximation constraint while the here proposed ADMM method does. Refer to Fig. 2 to see the dataset details.

However, the matrix $(\mathbf{A} - \mathbf{su}\mathbf{v}^T)$ is not necessarily nonnegative, presenting a serious obstacle to the overall approach.

Nonnegative Matrix *Underapproximation* (NMU) [7, 8, 9] appears as a solution to this challenge by posing the related problem

$$\min_{\mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n} \|\mathbf{A} - \mathbf{uv}^T\|_F^2 \quad \text{s.t.} \quad \begin{array}{l} \mathbf{u}, \mathbf{v} \geq \mathbf{0}, \\ \mathbf{A} \geq \mathbf{uv}^T. \end{array} \quad (\text{NMU})$$

Now, $\mathbf{A} - \mathbf{uv}^T$ is constrained to be a nonnegative matrix, rendering the previous iterative approach possible.

In this work, we first present a new efficient NMU technique, based on the Alternating Direction Method of Multipliers (ADMM).¹ The resulting algorithm is, to the best of our knowledge, the only NMU algorithm in the literature to efficiently solve Problem (NMU) while enforcing all its constraints (see Fig. 1). Additionally, it is easy to implement and delivers high-quality results.

We then present an application of the proposed NMU technique to the problem of robustly fitting multiple parametric models in a given dataset.¹ Fitting multiple instances of a given parametric model to data corrupted by noise and outliers is a widespread problem in computer vision. It is encountered in a diverse set of applications such as finding

¹Source code available at <https://goo.gl/xSqKQ4>.

lines/circles/ellipses in images, homography estimation in stereo vision, motion estimation/segmentation, and analysis of 3D point clouds. We draw the connection between NMU and this important problem in computer vision and pattern recognition and show that the proposed formulation leads (1) to NMU factors that are naturally interpretable and (2) to state-of-the-art results. This is also the case for an example on climate data here provided.

Organization. The remainder of this paper is organized as follows. In Sec. 2 we present the proposed NMU technique and provide an example pattern recognition application. In Sec. 3 we introduce the robust fitting formulation. We present computer vision experimental results in Sec. 4 and finally provide some closing remarks in Section 5.

Notation. Let \mathbf{X} be a matrix. $(\mathbf{X})_{ij}$, $(\mathbf{X})_{:j}$, $(\mathbf{X})_i$: denote the (i, j) -th entry of \mathbf{X} , the j -th column of \mathbf{X} , and the i -th row of \mathbf{X} , respectively.

2. Nonnegative Matrix Underapproximation

In several practical scenarios, the Lagrangian relaxation used in the considered state-of-the-art works [7, 8, 9] did not yield good results. In particular, this algorithm delivers factors \mathbf{u}, \mathbf{v} that violate the constraint $\mathbf{A} \geq \mathbf{u}\mathbf{v}^T$, defeating their own purpose and usefulness. We show such an example in Fig. 1.

In this work, we propose to use ADMM to solve Problem (NMU). In short, ADMM solves convex optimization problems by breaking them into smaller subproblems, which are individually easier to handle. It has also been extended to handle non-convex problems, e.g., to solve several flavors of NMF [5, 16, 17, 24].

Problem (NMU) can be equivalently re-formulated as

$$\min_{\substack{\mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n \\ \mathbf{R} \in \mathbb{R}^{m \times n}}} \frac{1}{2} \|\mathbf{R}\|_F^2 \quad \text{s.t.} \quad \begin{aligned} &\mathbf{R} = \mathbf{A} - \mathbf{u}\mathbf{v}^T, \\ &\mathbf{u}, \mathbf{v} \geq \mathbf{0}, \mathbf{R} \geq \mathbf{0}, \end{aligned} \quad (2)$$

and we consider its augmented Lagrangian,

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{R}, \mathbf{\Gamma}) = \frac{1}{2} \|\mathbf{R}\|_F^2 + \mathbf{\Gamma} \bullet (\mathbf{A} - \mathbf{u}\mathbf{v}^T - \mathbf{R}) + \frac{\gamma}{2} \|\mathbf{A} - \mathbf{u}\mathbf{v}^T - \mathbf{R}\|_F^2, \quad (3)$$

where $\mathbf{\Gamma} \in \mathbb{R}^{m \times n}$ is a Lagrange multiplier, γ is a penalty parameter, and $\mathbf{B} \bullet \mathbf{C} = \sum_{i,j} (\mathbf{B})_{ij} (\mathbf{C})_{ij}$ for matrices \mathbf{B}, \mathbf{C} of the same dimensions.

The ADMM algorithm works in a coordinate descent fashion, successively minimizing \mathcal{L} with respect to $\mathbf{u}, \mathbf{v}, \mathbf{R}$, one at a time while fixing the others at their most recent values, i.e.,

$$\mathbf{u}_k = \underset{\mathbf{u} \geq \mathbf{0}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}, \mathbf{v}_{k-1}, \mathbf{R}_{k-1}, \mathbf{\Gamma}_{k-1}), \quad (4a)$$

$$\mathbf{v}_k = \underset{\mathbf{v} \geq \mathbf{0}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}_k, \mathbf{v}, \mathbf{R}_{k-1}, \mathbf{\Gamma}_{k-1}), \quad (4b)$$

$$\mathbf{R}_k = \underset{\mathbf{R} \geq \mathbf{0}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}_k, \mathbf{v}_k, \mathbf{R}, \mathbf{\Gamma}_{k-1}), \quad (4c)$$

Algorithm 1: ADMM algorithm for NMU

input : nonnegative matrix \mathbf{A} , maximum number of iterations N_{iter} , convergence tolerance τ .

output: factor $\mathbf{u}_k, \mathbf{v}_k$.

```

1 Compute initial value for  $\mathbf{u}_0, \mathbf{v}_0$ ;
2  $\mathbf{\Gamma} \leftarrow \mathbf{0}$ ;
3  $\mathbf{R} \leftarrow \mathcal{P}_+(\mathbf{A} - \mathbf{u}_0\mathbf{v}_0^T)$ ;
4 foreach  $k \in \{1 \dots N_{\text{iter}}\}$  do
5    $\mathbf{u}_k \leftarrow \mathcal{P}_+((\mathbf{A} - \mathbf{R} + \mathbf{\Gamma}) \mathbf{v}_{k-1}^T)$ ;
6    $\mathbf{u}_k \leftarrow \mathbf{u}_k / \|\mathbf{u}_k\|_\infty$ ;
7    $\mathbf{v}_k \leftarrow \mathcal{P}_+(\mathbf{u}_k^T (\mathbf{A} - \mathbf{R} + \mathbf{\Gamma}))$ ;
8    $\mathbf{v}_k \leftarrow \mathbf{v}_k / (\mathbf{u}_k^T \mathbf{u}_k)$ ;
9    $\mathbf{R} \leftarrow \mathcal{P}_+(\mathbf{A} - \mathbf{u}_k\mathbf{v}_k^T + \mathbf{\Gamma})$ ;
10   $\mathbf{\Gamma} \leftarrow \mathbf{\Gamma} + \mathbf{A} - \mathbf{u}_k\mathbf{v}_k^T - \mathbf{R}$ ;
11  if  $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_2 / \|\mathbf{u}_{k-1}\|_2 < \tau$  and
     $\|\mathbf{v}_k - \mathbf{v}_{k-1}\|_2 / \|\mathbf{v}_{k-1}\|_2 < \tau$  then
12    return
```

and then updating the multiplier $\mathbf{\Gamma}_k$. For the problem at hand, each of these steps can be written in closed form as

$$\mathbf{u}_k = \mathcal{P}_+(\mathbf{M}\mathbf{v}_{k-1}^T / (\mathbf{v}_{k-1}^T \mathbf{v}_{k-1})), \quad (5a)$$

$$\mathbf{v}_k = \mathcal{P}_+(\mathbf{u}_k^T \mathbf{M} / (\mathbf{u}_k^T \mathbf{u}_k)), \quad (5b)$$

$$\mathbf{R}_k = \mathcal{P}_+(\mathbf{A} - \mathbf{u}_k\mathbf{v}_k^T + \gamma^{-1}\mathbf{\Gamma}_{k-1}), \quad (5c)$$

$$\mathbf{\Gamma}_k = \mathbf{\Gamma}_{k-1} + \xi\gamma(\mathbf{A} - \mathbf{u}_k\mathbf{v}_k^T - \mathbf{R}_k), \quad (5d)$$

where $\mathbf{M} = \mathbf{A} - \mathbf{R}_k + \mathbf{\Gamma}_{k-1}$ and for any matrix \mathbf{B}

$$(\mathcal{P}_+(\mathbf{B}))_{ij} = \max\{(\mathbf{B})_{ij}, 0\}. \quad (6)$$

In practice, we set $\gamma = 1$ and $\xi = 1$.

A complete pseudocode is provided in Alg. 1. As with most matrix factorization problems, there is a scaling indeterminacy in Problem (NMU), we can multiply and divide \mathbf{u} and \mathbf{v} respectively by a scalar constant and the result will remain unchanged. We thus arbitrarily set $\max(\mathbf{u}_k) = 1$ throughout the algorithm iterations and scale \mathbf{v}_k accordingly. The algorithm terminates when the relative change in both \mathbf{u} and \mathbf{v} is below a given tolerance τ .

Initialization. It is natural to use the rank-one SVD to initialize \mathbf{u}_0 and \mathbf{v}_0 . Let $\mathbf{x}, s, \mathbf{y}$ be the rank-one SVD of \mathbf{A} . Following our convention that $\max(\mathbf{u}_k) = 1$, we set

$$\mathbf{u}_0 = \mathbf{x} / \|\mathbf{x}\|_\infty, \quad \mathbf{v}_0 = \|\mathbf{x}\|_\infty s \mathbf{y}. \quad (7)$$

Extracting multiple NMU factors. As stated in the introduction, the algorithm for extracting multiple factors is straightforward. We iterate the following two steps:

1. extract rank-one NMU factors $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ from \mathbf{A} , and
2. $\mathbf{A} \leftarrow \mathbf{A} - \hat{\mathbf{u}}\hat{\mathbf{v}}^T$.

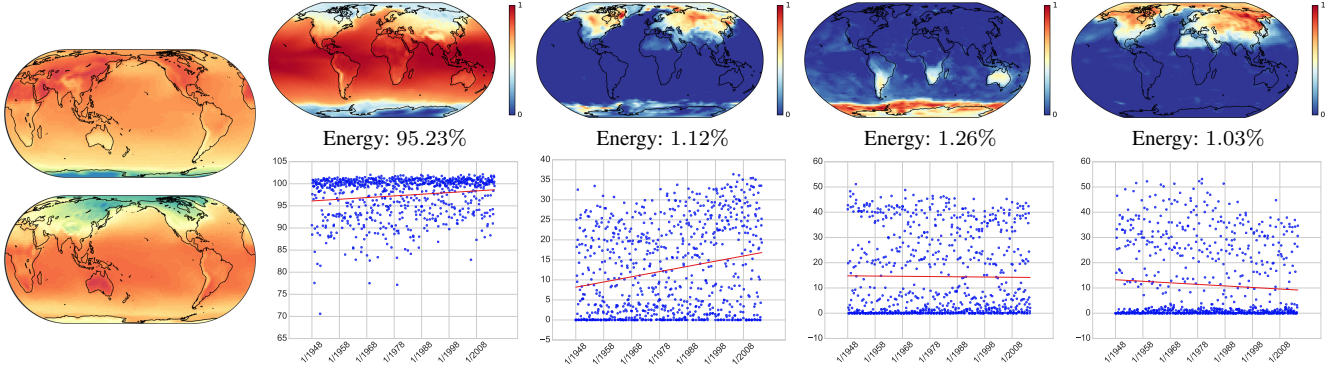


Figure 2. **NMU of climate data** (<http://www.esrl.noaa.gov/psd/repository/>). The data contains monthly mean surface temperatures arranged in a 144×73 grid since 1948 (800 months in total), forming a 10512×800 matrix. On the first column, we reproduce the NMF factors from [17], which represent the main seasonal effects (inverted winter and summer in the north and in the south). On the remaining columns, we show the first four NMU factors (left and right factors in the top and bottom rows, respectively), which amount to 98.64% of the matrix energy. Their patterns are very different from NMF, first extracting a main global component and then providing part-based insights on different areas. It is also interesting to note that an upward trend can be perceived in the loadings (right factors) of the first two components, something that is not perceived with NMF, see [17].

2.1. Illustrative application: Climate data analysis

NMF provides a very rich descriptive power for climate datasets [17]. The evidence of low rank models in climate data is of interest by itself, as they provide a concise way for describing the data. Nonnegativity is a useful addition since, under this model, the effects of different factors cannot cancel each other, providing models that are easily interpretable. Furthermore, the rank of the approximation can be estimated soundly in an online fashion.

In the first example we show that NMU provides radically different insights, when compared with NMF. The technical details and results of an experiment using climate data are shown in Fig. 2. Analyzing the NMU factors, we see that we first obtain one main global component, with nonzero entries both in the left and right factors. These factors are sparse in the remaining components (up to numerical precision), even if sparsity is not formally promoted.

In Fig. 3, we plot the relative errors of \mathbf{u}_k (left factor) and \mathbf{v}_k (right factor) for the first two components in Fig. 2. The algorithm converges very quickly to a steady solution. Let us point out that these are typical cases of the performance of the algorithm.

A keen observer might object that in Fig. 3 the convergence is much faster for the first pair of factors than for the second. This can be explained by the high energy of the NMU factors (95.23% of the total energy), making easier to extract coherent patterns (as these will be more evident).

3. Robust model fitting

We now illustrate how to use NMU and in particular the proposed ADMM algorithm for solving multi-model estimation problems in computer vision.

Finding a single parametric model instance is a robust

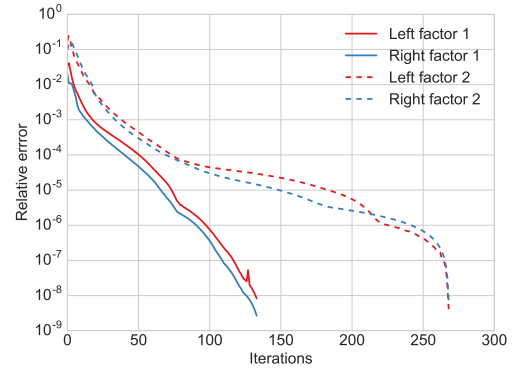


Figure 3. Convergence of Alg. 1 for the first two factors extracted from the example in Fig. 2. In both cases, the algorithm exhibits a very fast empirical convergence rate (notice the logarithmic scale).

fitting problem that is hard on its own. When an *unknown* number of instances might be present in the dataset the difficulty increases due to the unavoidable emergence of pseudo-outliers (data points that belong to one structure and are usually outliers to any other structure). Thus, we face the problem of simultaneous robust estimation of model parameters and attribution of data points to the estimated models. These two problems are intrinsically intertwined.

The given dataset $\{\mathbf{x}_i\}_{i=1}^m$ is composed of m geometric objects to which we seek to fit one or more multiple instances of a parametric model. Formally, a model μ is the zero level-set of a smooth parametric function $f_\mu(\mathbf{x}; \theta)$,

$$\mu(\theta) = \{\mathbf{x} \in \mathbb{R}^d, f_\mu(\mathbf{x}; \theta) = 0\}, \quad (8)$$

where θ is a parameter vector. The error associated with the datum $\mathbf{x} \in \mathcal{X}$ with respect to the model $\mu(\theta)$ is

$$e_\mu(\mathbf{x}, \theta) = \min_{\mathbf{x}' \in \mu(\theta)} \text{dist}(\mathbf{x}, \mathbf{x}'), \quad (9)$$

where dist is an appropriate distance function. We also define the soft-membership function as

$$\text{sm}_\mu(\mathbf{x}, \theta) = \begin{cases} \exp(-d^2/(2\sigma^2)) & \text{if } d \leq 3\sigma; \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

where $d = e_\mu(\mathbf{x}, \theta)$ and σ is the main (and only) parameter of the proposed approach.

The input of the algorithm is a collection $\mathcal{U} = \{\theta_j\}_{j=1}^n$ of model parameters, obtained with a RANSAC-like sampling technique. There are multiple methods in the literature to obtain such a collection, e.g., [6, 22, 23]. In Alg. 2 we show a basic prototypical example. The $m \times n$ preference matrix \mathbf{P} , whose rows and columns represent respectively the m = data elements $\{\mathbf{x}_i\}_{i=1}^m$ and the $n = |\mathcal{U}|$ models, with entries defined as

$$(\mathbf{P})_{ij} = \text{sm}_\mu(\mathbf{x}_i, \theta_j). \quad (11)$$

See Fig. 4 for a preference matrix example.

Following [16, 18], we pose the problem of robustly fitting multiple parametric models as a biclustering problem. In other words, we look for clusters in the space defined by Cartesian product of elements and models. In this case, the iteratively computed NMU factors are the extracted biclusters. As we will see in the following, using NMU in this context leads to an intuitive and simple interpretation of the retrieved biclusters.

Initialization. In our robust fitting experiments, we found that initializing Alg. 1 with a rank-one SVD does not provide optimal results. Perhaps unsurprisingly, the strategy chosen by RANSAC (selecting the largest consensus set) proved to be a very good initial condition to our NMU algorithm. We thus pick the largest column of \mathbf{P} , i.e., $j^* = \arg\max_j \|(\mathbf{P})_{:j}\|_1$, and set

$$\mathbf{u}_0 = (\mathbf{P})_{:j^*} / \|(\mathbf{P})_{:j^*}\|_\infty, \quad (12a)$$

$$\mathbf{v}_0 = \|(\mathbf{P})_{:j^*}\|_\infty \mathbf{u}_0^T \mathbf{P} / (\mathbf{u}_0^T \mathbf{u}_0). \quad (12b)$$

Extracting multiple NMU factors from \mathbf{P} . This is a particular example of biclustering, where we know that each column of \mathbf{P} corresponds to *at most one* ground truth model. In other terms, it does not make sense to describe any given column by the combination of multiple models/biclusters. In practice, this means that once a column $(\mathbf{P})_{:j}$ is encoded in an NMU factor (via a positive load $(\mathbf{v})_j$), we can remove it entirely from \mathbf{P} . Hence, instead of subtracting $\mathbf{u}\mathbf{v}^T$ from \mathbf{P} , we simply set the columns with positive loads to zero.

3.1. Interpretation of the NMU factors

We consider the solution $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ to Problem (NMU), provided by Alg. 1, with \mathbf{P} as the input matrix. In general terms, $(\hat{\mathbf{u}})_i, (\hat{\mathbf{v}})_j$ indicate the soft-membership of element

Algorithm 2: Random sampling algorithm

input : set of objects \mathcal{X} , parametric function f_μ .
output: pool \mathcal{U} of consensus sets.
1 $b \leftarrow$ minimum number of elements necessary to uniquely characterize model μ , see Eq. (8);
2 **foreach** $j \in \{1 \dots n\}$ **do**
3 Select at random a set \mathcal{X}_j of b elements from \mathcal{X} ;
4 Estimate θ_j from \mathcal{X}_j ;
5 $\mathcal{U} \leftarrow \{\theta_j\}_{j=1}^n$;

\mathbf{x}_i to model θ_j . If the NMU factorization considers that $(\mathbf{P})_{:j}$ “agrees” with $\hat{\mathbf{u}}$, then $(\hat{\mathbf{v}})_j > 0$; otherwise $(\hat{\mathbf{v}})_j = 0$.

Let us now analyze the underapproximation constraint. We first point out that since we minimize $\|\mathbf{A} - \mathbf{u}\mathbf{v}^T\|_F$, the underapproximation is as tight as possible. Now, for any column $(\mathbf{P})_{:j}$, we have that $(\mathbf{P})_{:j} \geq \hat{\mathbf{u}}(\hat{\mathbf{v}})_j$. Let \mathbf{d}_j be the m -dimensional vector such that $(\mathbf{d}_j)_i = e_\mu(\mathbf{x}_i, \theta_j)$. Since $0 \leq (\mathbf{P})_{ij} \leq 1$, $0 \leq \hat{\mathbf{u}}, \hat{\mathbf{v}} \leq 1$; hence,

$$-\log(\mathbf{P})_{:j} \leq -\log \hat{\mathbf{u}} - \log(\hat{\mathbf{v}})_j, \quad (13a)$$

$$\frac{1}{2}(\mathbf{d}_j/\sigma)^2 \leq -\log \hat{\mathbf{u}} - \log(\hat{\mathbf{v}})_j, \quad (13b)$$

$$\mathbf{d}_j \leq -\sigma \log^{1/2} \hat{\mathbf{u}}^2 - \sigma \log^{1/2}(\hat{\mathbf{v}})_j^2 \quad (13c)$$

where the logarithm is taken element-wise. The nonnegative scalar $-\sigma \log^{1/2}(\hat{\mathbf{v}})_j^2$ provides a global safety margin over the distances \mathbf{d}_j to ensure that they are included in the solution. The higher the agreement, the lower the margin (if $(\hat{\mathbf{v}})_j = 0$, we have an infinite margin). A similar pattern is observed for $-\sigma \log^{1/2} \hat{\mathbf{u}}^2$, in this case individualizing the margin for each data point \mathbf{x}_i (whether more or less leniency is needed to include it in the solution).

This interpretation of the values of $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$, being derived from the underapproximation constraint, is not present in other formulations of the robust fitting problem [3, 16].

3.2. Computing model instances from NMU factors

Once the NMU factors $\hat{\mathbf{u}}_t, \hat{\mathbf{v}}_t$ are found, we estimate the corresponding $\hat{\theta}_t$ by weighted least-squares regression, i.e.,

$$\hat{\theta}_t = \arg\min_{\theta} \sum_{i=1}^m (\hat{\mathbf{u}}_t)_i [e_\mu(\mathbf{x}_i, \theta)]^2. \quad (14)$$

Each data point contributes to the estimation of the final model instance proportionally to its soft-membership to the factor $\hat{\mathbf{u}}$.

3.3. Filtering candidates with statistical testing

Not all of the extracted NMU factors will correspond to meaningful models; in particular, the very last factors will correspond to noise in \mathbf{P} (i.e., spurious models). We use

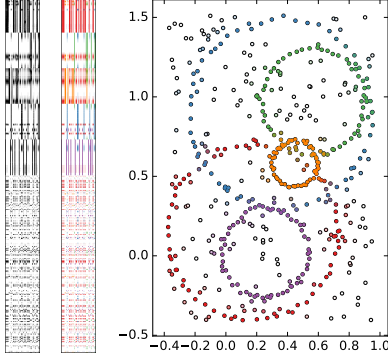


Figure 4. Preference matrix (Eq. (11)), before and after factorization (each NMU factors shown with a different color). We also show the computed soft-membership of points to the detected models. Points at the intersection of two circles are correctly assigned to both circles.

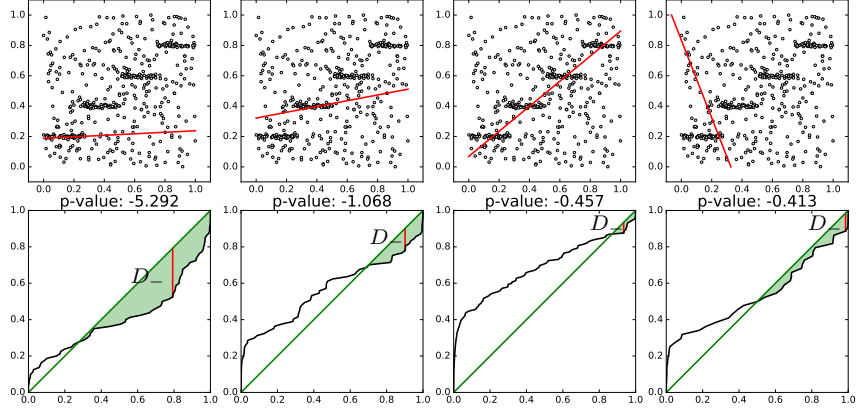
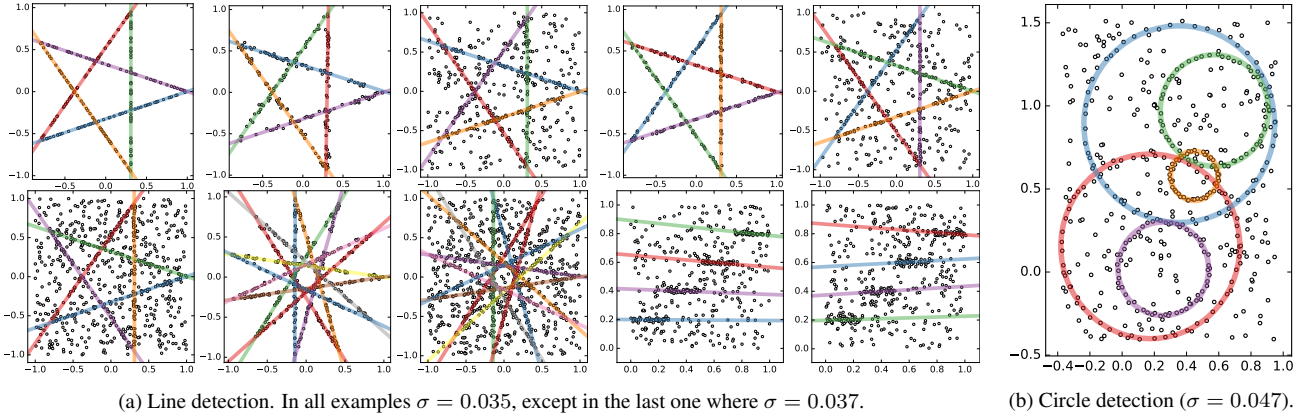


Figure 5. Examples of the statistical test presented in Sec. 3.3 with $\sigma = 0.015$ (empirical CDFs shown in the bottom row). The p-values are given in a base-10 logarithmic scale. When there is a higher concentration of points near the line than in the surrounding region (leftmost example), the p-value becomes very small. The p-value grows quickly as the line deviates from such a setting.



(a) Line detection. In all examples $\sigma = 0.035$, except in the last one where $\sigma = 0.037$.

(b) Circle detection ($\sigma = 0.047$).

Figure 6. Results on synthetic 2D datasets [19]. We show with different colors the final models estimated from the extracted biclusters (some colors may repeat themselves). In every case, the models are correctly retrieved. The lines in the last example of Fig. 6(a) are noisier and, as one might expect, a higher σ is needed to recover them. Estimating a circle from 3 points is much noisier than estimating a line from 2 points, as the curvature might change dramatically; hence, a higher σ is also needed in this case.

statistical testing to determine whether each NMU factor $\hat{\mathbf{u}}_t, \hat{\mathbf{v}}_t$ is to be kept in the final result.

Let F_t be the empirical cumulative distribution function (CDF) of the values $\text{sm}_\mu(\mathbf{x}_i, \hat{\theta}_t)$ and F_U denote the CDF for the uniform distribution over the interval $[0, 1]$. Following Kuiper's statistical test, we define the statistic

$$D_- = \max_{0 \leq x \leq 1} F_U(x) - F_t(x). \quad (15)$$

Since Kuiper's test is closely related to the Kolmogorov-Smirnov test, the statistic D_- follows a Kolmogorov-Smirnov CDF. In Fig. 5, we can see several examples of its value for different lines. The p-value of D_- is a good indicator of a high concentration of values near one. Hence, it is a good indicator of a high concentration of elements near the evaluated instance model.

The null hypothesis is rejected if $D_- > K_\alpha$ where K_α is found from $\Pr(K > K_\alpha) = \alpha$. The value α is automat-

ically set to $1/\binom{m}{b}$ where $\binom{m}{b}$ is the combinatorial number and b is the minimum number of elements necessary to estimate/instantiate a model [4].

This test can also be used as a preprocessing step. The standard random sampling approach to multiple model estimation generates many good model instances (composed of inliers), but also generates many bad models (composed mostly of outliers). Then, prior to computing the NMU factors, we discard the columns of $(\mathbf{P})_{:,j}$ for which the null hypothesis is not rejected. Here, we also set $\alpha = 1/\binom{m}{b}$.

3.4. From NMU factors to independent models

In our framework, models are allowed to share elements as there are no orthogonality constraints between successive NMU factors. However, if two distinct factors $\hat{\mathbf{u}}_t$ and $\hat{\mathbf{u}}_{t'}$ are too similar, the estimated models $\hat{\theta}_t$ and $\hat{\theta}_{t'}$ will be too similar, producing redundant output information.

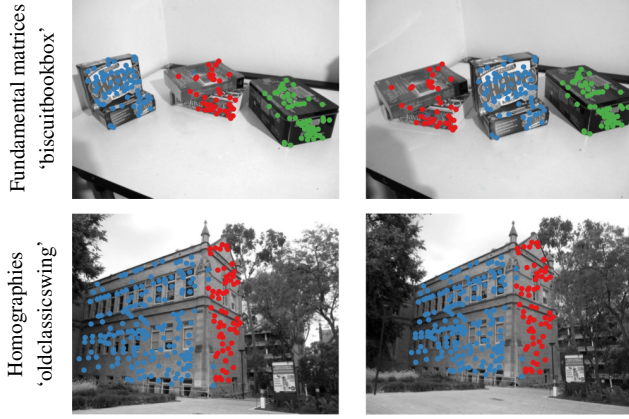


Figure 7. Two prototypical examples of the results in Tab. 1 with misclassification errors 1.16% (top) and 1.32% (bottom).

To avoid this, we compute the correlation $c_{t,t'}$ between every pair of factors t, t' . We consider two factors independent from each other if this quantity is lower than 0.6 (i.e., an angle between them is above 53° approx., which seems to a reasonable universal choice). We build a graph by linking together the vertices t and t' if $c_{t,t'} > 0.6$. Notice that the number of factors is a very small number here, making the graph small. We then list all maximal independent sets (MIS) in this graph. Each MIS is a set of mutually compatible factors/models.

We seek to determine which MIS of models/factors is the one better describing the data. Many different techniques can be used for this task, such as the Bayesian and Akaike information criteria or minimum description length. We follow a simplified approach: for each MIS, we consider the geometric mean of the p-values associated with its models (computed with the previously described statistical test); we return the MIS with minimum geometric mean. We note that results could be improved by using a more sophisticated selection method.

Optionally and if required by the application, as the last step of our algorithmic pipeline, we can force the models to have an empty intersection. There are many alternative ways to address this assignment problem. In this work, we simply assign elements in the intersection of several models to the closest model in distance (see Eq. (9)).

4. Experimental results

We start our experimental evaluation with a few small synthetic datasets [19] where 2D points are arranged forming lines and circles. The results are shown in Fig. 6, where all models are clearly detected. In this case, we use Alg. 2 as a sampling strategy, all points being treated as equiprobable (we sample from a uniform distribution).

The proposed approach can correctly recover overlap-

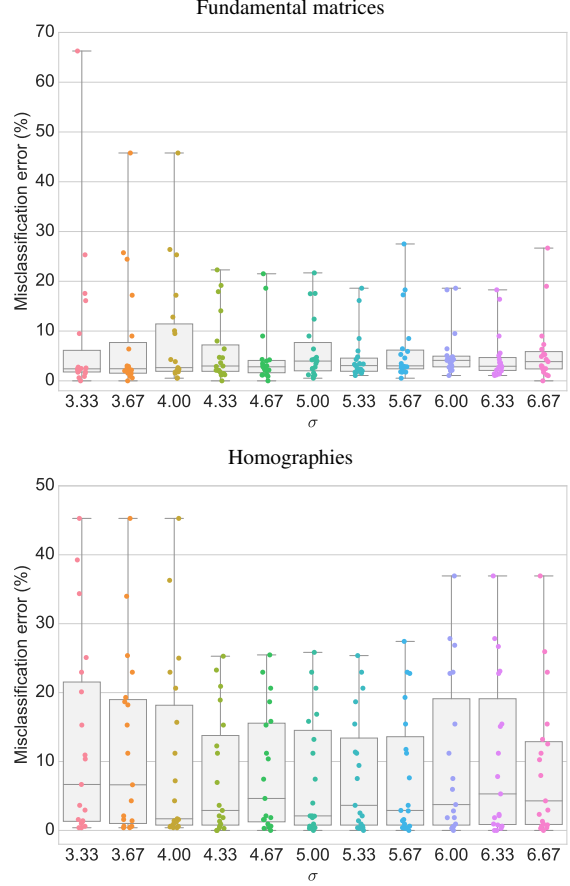


Figure 8. Results on the AdelaideRMF dataset [22] for different σ (Eq. (10)). A small horizontal jitter was added to each colored point in the plot to improve visualization. The results are roughly stable to the choice of σ , making this selection less critical than in the hard-membership approaches, i.e., with a binary preference matrix, e.g. [18, 19].

ping models, see Fig. 6. This can be observed in detail by comparing the results in Fig. 6(b) with its preference matrix in Fig. 4. This is an intrinsic limitation of previous state-of-the-art competitors such as J-linkage [19], T-linkage [10], RPA [11] and most parametric fitting techniques, which are generally based on partitioning (clustering) the dataset.

We next estimate multiple fundamental matrices (moving camera and moving objects) and multiple homographies (moving planar objects) on the images in the AdelaideRMF dataset [22]. This is a standard dataset in the literature (e.g., [3, 10, 11, 15, 18, 22]), but as pointed out in [18] it contains a non-negligible quantity of errors in its ground truth. This implies that, above some point, an improvement in the actual performance might not necessarily reflect itself as an improvement compared to the ground truth.

As standard in the literature [3, 10, 11, 15, 18] we use the misclassification error (ratio of misclassified points) to measure performance. While the ground truth assigns elements to groups in a hard fashion, the values $sm_\mu(\mathbf{x}_i, \hat{\theta}_t)$

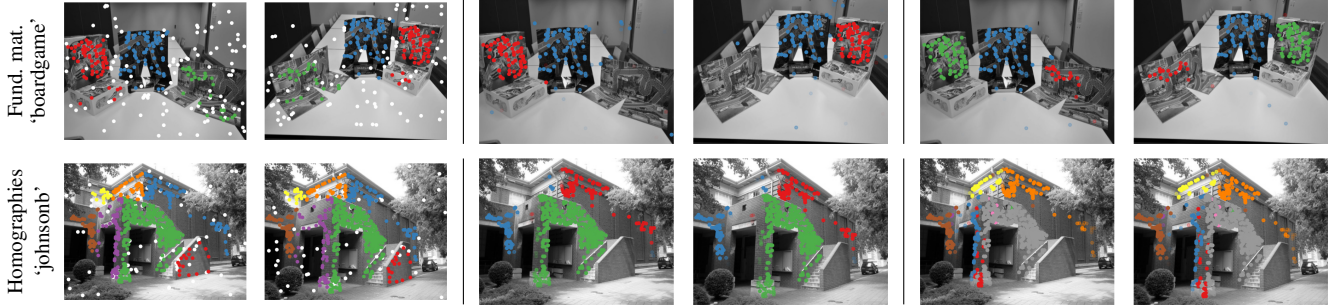


Figure 9. Two examples in which the method exhibits a higher misclassification error (ME) in Tab. 1. From left to right: ground truth hard assignments, RS-NMU soft assignments, and RS-NMU soft assignments with no statistical testing performed (in pre- nor in post-processing). Top row: the ME goes from 21.51% (center) to 10.75% (right); the main difference is the appearance of an additional model that had been discarded for not passing the statistical test. Bottom row: the ME goes from 23.27% (center) to 15.56% (right); a few ‘desired’ groups appear (blue and yellow dots), but accompanied with some ‘undesired’ groups (red and pink dots). The problem of differentiating the ‘desired’ from the ‘undesired’ groups is a very hard when they contain few points.

are continuous in $[0, 1]$. To compute this error, we simply binarize them by considering that an element \mathbf{x}_i belongs to group t if $\text{sm}_\mu(\mathbf{x}_i, \hat{\theta}_t) > 0$.

To show that the proposed method is agnostic to the random sampling technique, we use MultiGS [2] for these experiments.² Of the total running time, MultiGS uses on average 98%, while the remaining 2% corresponds to the proposed ADMM algorithm for NMU.

The results on the full dataset are presented in Tab. 1. For this experiment, we term the approach proposed in Sec. 3 as RS-NMU. We clarify that RPA [11] and FABIA [3] achieve good results; however, to yield these performances, they use the *ground truth* number of models as an input. The remaining methods, as well as the method here proposed, automatically estimate this number.

To estimate fundamental matrices, see Tab. 1(a), RS-NMU outperforms all its competitors, even being slightly better on average than RPA/FABIA. To estimate homographies, see Tab. 1(b), the differences with other methods are even higher, with RS-NMU clearly outperforming them. These observations hold true, even if the σ is not accurately set (compare the last two columns in tabs. 1(a) and 1(b)). In Fig. 7 we show two examples of the RS-NMU results. This improved performance comes from a lightweight algorithm, using approx. 2% of the total running time.

As one might expect, there is not a universal value of σ that works best for all the image pairs, see for example ‘breadcube’ and ‘breadcube chips’ in Tab. 1(a).

We analyze in Fig. 8 the performance of RS-NMU as σ changes. We can see that the performance is not dramatically affected, providing to the user assurance that there is no need to fine-tune σ in order to obtain good results.

In Fig. 9, we present two image pairs for which RS-NMU

presents sub-optimal results. We show with a simple argument that these results are not due to the NMU approach nor to the proposed ADMM algorithm. The group of green points in the top-left ground truth pair has errors, as pointed out in [18]. This means that the effective size of this group is smaller than what is marked in the ground truth. Thus, it does not pass the statistical test in Sec. 3.3. If we disable testing in our pipeline, we recover the error-less part of this group. However, disabling the testing might lead to recover other spurious groups as well, see the bottom row of Fig. 9.

5. Conclusions

In this work, we first presented a highly efficient algorithm to address the nonnegative matrix underapproximation (NMU) problem. The solutions provided by our algorithm are the only ones in the literature for which the underapproximation constraint holds. NMU’s results are interesting as, compared to traditional NMF, they present additional sparsity and part-based behavior without explicitly adding sparsity-inducing priors. To show this, we have presented a practical application to the analysis of climate data.

We have also presented an application to the task of robustly fitting multiple parametric models to a dataset. Accompanied by a specially designed algorithmic pipeline, NMU delivers state-of-the-art results, outperforming other alternatives in the literature.

In the future, we look forward to applying our NMU algorithm to the analysis of other datasets, since it will provide alternative insights to the ones obtained with other matrix factorization tools.

Finally, to obtain a completely autonomous and parameterless robust fitting system, we need to automatically estimate σ , not only for each image pair, but for each particular model, as different models might have different noise levels. An interesting lead would be to use the statistical test in Sec. 3.3 as a way to infer the optimal σ by computing the model’s optimal p-value.

²Interestingly, a recent technique [1] has reduced to two the minimal sample size (see Alg. 2) for fundamental matrices; its use would significantly reduce the computational complexity of the sampling step, imposing an upper bound of $O(m^2)$ to the total number of possible combinations.

	RSE [18]	ARSE [18]	T-linkage [10]	RCMSA [15]	RPA [11]	FABIA [3]	RS-NMU $\sigma = 4.67$	RS-NMU $\sigma = 5.33$
biscuit	22.60	0.9	16.93	14.00	1.15	0.00	0.91	2.42
biscuitbook	3.50	1.2	3.23	8.41	3.23	1.32	2.05	2.35
biscuitbookbox	3.90	1.90	3.10	16.92	3.88	3.86	1.16	1.54
boardgame	15.4	15	21.43	19.80	11.65	8.96	21.51	16.13
book	3.80	2.20	3.24	4.32	2.88	0.00	0	1.07
breadcartoychips	27.70	28.60	14.29	25.69	7.50	4.22	4.22	3.38
breadcube	2.60	0.9	19.31	9.87	4.58	19.42	2.48	3.31
breadcubechips	15.70	14.80	3.48	8.12	5.07	0.87	2.17	1.74
breadtoy	1.4	2.2	5.40	3.96	2.76	19.62	3.47	3.47
breadtoycar	25.00	25.00	9.15	18.29	7.52	0.60	3.01	4.82
carchipscube	12.80	12.80	4.27	18.90	6.50	1.52	1.21	1.82
cube	2.4	12.50	7.80	8.14	3.28	1.66	3.97	4.30
cubebreadtoychips	13.70	14.30	9.24	13.27	4.99	1.07	3.06	3.06
cubechips	2.90	3.60	6.14	7.70	4.57	0.53	2.82	2.46
cubetoy	3.80	4.60	3.77	5.86	4.04	2.21	1.2	2.01
dinobooks	23.60	18.3	20.94	23.50	15.14	9.72	18.61	18.61
game	0.4	0.9	1.30	5.07	3.62	0.00	4.29	6.01
gamebiscuit	9.90	4.30	9.26	9.37	2.57	2.44	2.13	1.83
toycubecar	8.1	8.1	15.66	13.81	9.43	9.50	9.00	8.50
Mean	10.48	9.06	9.37	12.37	5.49	4.61	4.59	4.68
STD	8.97	8.57	6.66	6.58	3.48	6.13	5.79	4.83
Median	8.10	4.60	7.80	9.87	4.57	1.66	2.82	3.06

(a) Misclassification error (%) for the estimation of multiple fundamental matrices.

	RSE [18]	ARSE [18]	T-linkage [10]	RCMSA [15]	RPA [11]	FABIA [3]	RS-NMU $\sigma = 4.33$	RS-NMU $\sigma = 5.00$
barrsmith	3.4	4.7	49.79	20.14	36.31	29.88	11.20	11.20
bonhall	29.20	38.00	21.84	19.69	41.67	24.02	25.28	25.84
bonython	1.60	5.20	11.92	17.79	15.89	6.82	0	0
elderhalla	6.10	3.70	10.75	29.28	0.93	3.04	0.93	0.93
elderhallb	12.2	20.40	31.02	35.78	17.82	18.63	15.29	16.86
hartley	1.60	1.30	21.90	37.78	17.78	23.59	0.62	0.62
johnsona	13.9	16.4	34.28	36.73	10.76	17.96	20.91	20.64
johnsonb	15.1	17.60	24.04	16.46	26.76	24.50	23.27	22.96
ladyssymon	3.50	2.60	24.67	39.50	24.67	11.81	2.11	2.11
library	2.40	0.9	24.53	40.72	31.29	20.47	1.86	1.86
napiera	8.20	11.00	28.08	31.16	9.25	21.85	3.64	3.97
napierb	18.1	18.60	13.5	29.40	31.22	36.68	18.92	15.83
neem	4.30	7.80	25.65	41.45	19.86	11.20	2.9	2.07
nese	1.20	2.10	7.05	46.34	0.83	4.92	0	0.39
oldclassicswing	1.7	2.50	20.66	21.30	25.25	7.92	1.32	2.11
physics	28.20	22.30	29.13	48.87	0.00	0.00	12.26	13.21
sene	0.40	0.80	7.63	20.20	0.42	2.20	0.4	0.4
unihouse	7.60	11.20	33.13	2.56	5.21	15.76	6.96	7.44
unionhouse	0.90	0.90	48.99	2.64	10.87	21.54	0.3	0.3
Mean	8.40	9.89	24.66	28.30	17.20	15.94	7.8	7.83
STD	8.90	10.04	11.96	13.45	12.87	10.10	8.85	8.76
Median	4.30	5.20	24.53	29.40	17.78	17.96	2.9	2.11

(b) Misclassification error (%) for the estimation of multiple homographies.

Table 1. Results on the AdelaideRMF dataset [22]. RPA/FABIA do not automatically determine the number of models (they draw this number from the ground truth). Grayed cells mark the first (dark gray) and second (light gray) best results among the methods that automatically estimate this number.

References

- [1] G. Ben-Artzi, T. Halperin, M. Werman, and S. Peleg. Two points fundamental matrix. Technical report, 2016. [7](#)
- [2] T. J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):625–638, 2012. [7](#)
- [3] M. Denitto, L. Magri, A. Farinelli, A. Fusiello, and M. Bicego. Multiple structure recovery via probabilistic bi-clustering. In *SSPR*, 2016. [4](#), [6](#), [7](#), [8](#)
- [4] A. Desolneux, L. Moisan, and J. M. Morel. *From Gestalt Theory to Image Analysis*, volume 34. Springer-Verlag, 2008. [5](#)
- [5] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the β -divergence. *Neural Computation*, 23(9):2421–2456, 2011. [2](#)
- [6] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. [4](#)
- [7] N. Gillis and F. Glineur. Using underapproximations for sparse nonnegative matrix factorization. *Pattern Recognition*, 43(4):1676–1687, 2010. [1](#), [2](#)
- [8] N. Gillis and R. J. Plemmons. Dimensionality reduction, classification, and spectral mixture analysis using non-negative underapproximation. *Optical Engineering*, 50(2):027001, 2011. [1](#), [2](#)
- [9] N. Gillis and R. J. Plemmons. Sparse nonnegative matrix underapproximation and its application to hyperspectral image analysis. *Linear Algebra and Its Applications*, 438(10):3991–4007, 2013. [1](#), [2](#)
- [10] L. Magri and A. Fusiello. T-linkage: A continuous relaxation of J-linkage for multi-model fitting. In *CVPR*, pages 3954–3961, 2014. [6](#), [8](#)
- [11] L. Magri and A. Fusiello. Robust multiple model fitting with preference analysis and low-rank approximation. In *BMVC*, pages 20.1–20.12, 2015. [6](#), [7](#), [8](#)
- [12] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000. [1](#)
- [13] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. [1](#)
- [14] E. E. Papalexakis, N. D. Sidiropoulos, and R. Bro. From K-means to higher-way co-clustering: multilinear decomposition with sparse latent factors. *IEEE Trans. Signal Process.*, 61(2):493–506, 2013. [1](#)
- [15] T. T. Pham, T. J. Chin, J. Yu, and D. Suter. The random cluster model for robust geometric fitting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1658–1671, 2014. [6](#), [8](#)
- [16] M. Tepper and G. Sapiro. A bi-clustering framework for consensus problems. *SIAM J. Imaging Sci.*, 7(4):2488–2525, 2014. [2](#), [4](#)
- [17] M. Tepper and G. Sapiro. Compressed nonnegative matrix factorization is fast and accurate. *IEEE Trans. Signal Process.*, 64(9):2269–2283, 2016. [2](#), [3](#)
- [18] M. Tepper and G. Sapiro. Fast L1-NMF for Multiple Parametric Model Estimation. Technical report, 2016. [4](#), [6](#), [7](#), [8](#)
- [19] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *ECCV*, 2008. [5](#), [6](#)
- [20] S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2010. [1](#)
- [21] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009. [1](#)
- [22] H. S. Wong, T. J. Chin, J. Yu, and D. Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *ICCV*, pages 1044–1051, 2011. [4](#), [6](#), [8](#)
- [23] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognit. Lett.*, 11(5):331–338, 1990. [4](#)
- [24] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Front. Math. China*, 7(2):365–384, 2012. [2](#)